

Análisis y Evaluación del MDD (Model Driven software Development) desde la Perspectiva del Nivel 2 del CMMI-DEV 1.3

Viviana Esterkin¹, Claudia Pons¹

⁽¹⁾Centro de Altos Estudios en Tecnología Informática (CAETI)
Av. Montes de Oca 745. Ciudad Autónoma de Buenos Aires
vesterkin@gmail.com, Claudia.pons@uai.edu.ar

Abstract. Los investigadores de la comunidad MDD sostienen que el Administrador de un proyecto MDD debe administrar 2 proyectos en simultáneo: el proyecto interno de herramientas MDD y el proyecto externo del aplicativo de negocios. Si una Organización quiere certificar CMMI nivel 2, deberá certificar ambos proyectos. En este trabajo, analizamos únicamente la certificación CMMI nivel 2 del proyecto interno MDD. Para ello, se han seleccionado buenas prácticas MDD propuestas en la literatura y basándose en ellas, se analiza si MDD brinda soporte a las prácticas específicas definidas por CMMI dev 1.3. Se concluye cuáles son las Áreas de Proceso CMMI nivel 2, que soporta MDD y se analiza el grado de este soporte.

Keywords: Capability Maturity Model (CMMI-DEV 1.3), Model Driven software Development(MDD), Area de Proceso.

1. Introducción

Los integrantes de la comunidad informática reconocen ampliamente los problemas que aparecen en el desarrollo de software en las empresas, por ejemplo, la codificación que implica una tarea intensiva, la aparición de nuevas tecnologías que obliga a rehacer el trabajo una y otra vez y el continuo cambio de los requerimientos debido a las necesidades del negocio y de las tecnologías. MDD (en inglés, Model Driven Development) representa un enfoque, definido por el Object Management Group (OMG) [5], para resolver algunos de estos problemas y para la mejora de las prácticas actuales de la ingeniería de software. En el sitio del OMG [5], pueden encontrarse ejemplos de casos exitosos de introducción del MDD en diferentes organizaciones.

Actualmente, la certificación de calidad de las empresas es una necesidad, porque la aplicación de las normas reconocidas internacionalmente, hacen que mejoren su productividad, incrementen su competitividad, además de posicionar a la compañía en el mercado externo. El modelo de calidad CMMI (Capability Maturity Model for Integration) [4], del Software Engineering Institute de Carnegie Mellon es uno de los más reconocidos internacionalmente y es el que utilizaremos en este trabajo.

Los investigadores ([1], [2], [3]), coinciden en afirmar que, cuando se implementa un proyecto MDD, el administrador del proyecto debe dirigir dos proyectos: el proyecto de herramientas MDD y el proyecto de negocios. Bajo este entorno, si una

organización desarrolla un proyecto MDD y quiere certificar un determinado nivel CMMI, los dos proyectos deberán certificar CMMI. En este trabajo, nos dedicaremos al proyecto de herramientas MDD únicamente y evaluaremos si las prácticas MDD, soportan el nivel de madurez 2 del CMMI.

Este estudio, está organizado del siguiente modo: el presente capítulo 1 es la Introducción, el capítulo 2 se refiere a los conceptos más importantes del MDD, el capítulo 3 resume algunos conceptos de CMMI, el capítulo 4 introduce las buenas prácticas de MDD seleccionadas y que serán utilizadas para evaluar si MDD soporta y en qué medida el nivel 2 de CMMI, el capítulo 5, analiza las áreas de proceso de CMMI nivel 2 y discute en qué medida las soporta MDD, y el capítulo 6 analiza los resultados obteniendo las conclusiones.

2. Desarrollo Dirigido por Modelos

El desarrollo dirigido por modelos es un enfoque para el desarrollo de software en el que los artefactos primarios de software son modelos, a partir de los cuales se genera el código y otros artefactos [2]. MDD propone resolver los problemas actuales de desarrollo de software, utilizando un marco de trabajo que asegura, portabilidad, interoperabilidad entre plataformas, independencia de plataforma y productividad [6].

Las cualidades de mayor importancia de los modelos deben ser: su comprensibilidad, su precisión para representar adecuadamente la realidad, su consistencia, su completitud, su modificabilidad, su reusabilidad y su posibilidad de validación y verificación [1]. Los elementos fundamentales para poner en práctica el proceso MDD son los modelos, los lenguajes estándar para representarlos, las transformaciones de modelos, los lenguajes para expresar las transformaciones de modelos, las herramientas de software que dan soporte a la creación de modelos como los editores de modelos, los simuladores de modelos, repositorios de modelos y transformaciones y otros artefactos[1].

3. Capability Maturity Model Integration (CMMI)

Los modelos CMMI, del Software Engineering Institute de la Universidad de Ingeniería de Software de Carnegie Mellon [7], son colecciones de mejores prácticas. CMMI, es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software y provee una guía para aplicar las mejores prácticas CMMI a estos procesos. En este estudio, se utilizará CMMI para desarrollo versión 1.3 (en inglés CMMI for Development, CMMI-DEV 1.3) [4].CMMI tiene cinco niveles de madurez, que indican cada uno, el nivel de madurez al que ha llegado una organización en el desarrollo de los procesos de software.

Un Area de Proceso, es un conjunto de prácticas relacionadas de dicha área que, cuando se implementan colectivamente, satisfacen un conjunto de objetivos que se consideran importantes para lograr su mejora. Los niveles de madurez, se utilizan para describir un camino de evolución, recomendado a una organización que desea mejorar los procesos que utiliza para desarrollar productos o servicios. Los cinco

niveles de madurez son: 1.Inicial, 2.Gerenciado, 3.Definido, 4.Gerenciado Cuantitativamente y 5.Optimizado. Los niveles de madurez se utilizan para caracterizar la mejora de una organización relativa a un conjunto de áreas de proceso.

En este trabajo, evaluaremos si MDD soporta el nivel de madurez 2 del CMMI, por lo que analizaremos las 7 Áreas de Proceso de nivel 2, que son, en orden alfabético, de acuerdo a [4], las siguientes: **Gestión de Configuraciones** (en inglés, Configuration Management, CM), **Gestión de los Acuerdos con Proveedores** (en inglés, Supplier Agreement Management, SAM), **Gestión de los Requerimientos** (en inglés, Requirements Management, REQM), **Aseguramiento de la Calidad del Proceso y del Producto** (en inglés, Process and Product Quality Assurance, PPQA), **Medición y Análisis** (en inglés, Measurement and Analysis, MA), **Monitoreo y Control del Proyecto** (en inglés, Project Monitoring and Control, PMC), **Planeamiento del Proyecto** (en inglés, Project Planning, PP). Cada área de procesos se describe en términos de prácticas específicas, que al implementarse, conducen a satisfacer sus objetivos.

4. Las Buenas Prácticas MDD

En este trabajo, evaluaremos, a partir de las buenas prácticas MDD, el soporte al nivel madurez 2 del CMMI. Para seleccionar las buenas prácticas del MDD, se han tenido en cuenta, los puntos de vista de tres autores especialistas en el tema, que son Claudia Pons et al [1], Peter Swithinbank et al [2] y Erkuden Ríos et al [3]. En este capítulo, se enuncian los criterios que se han elegido de cada uno de ellos.

Se identifica cada práctica seleccionada, con un número (comenzando por el número 1 para cada autor) y una letra que indica su autor/es o procedencia (en el caso del Red Book de IBM). Para Erkuden Ríos et al [3], se mantiene a continuación de lo indicado anteriormente, la codificación utilizada por los autores para una mejor identificación.

A continuación se pasa a enumerar las prácticas seleccionadas.

4.1 Swithinbank et al [2]

De acuerdo al capítulo 4 del Red Book de IBM [2], en el proceso MDD se deben realizar las siguientes tareas:

- 1R. Identificar Patrones Comunes y estándares.
- 2R. Identificar los activos MDD existentes que pueden reusarse
- 2Ra. Considerar si las herramientas MDD que van a generarse serán reusables.
- 3R. Definir el modelo de diseño
- 4R. Identificar el modelo UML de componentes, independiente del tiempo de ejecución.
- 5R. Producir los Artefactos de Muestra para los escenarios clave
- 6R. Definir la cadena de herramientas
- 6.1R. Extraer Templates a partir de los Artefactos de Muestra
- 6.2R. Diseñar, codificar y testear las transformaciones
- 6.2Ra. Validar la cadena de herramientas usando los escenarios clave
- 6.3R. Producir documentación y educación para los desarrolladores
- 6.3Ra. Entrenar a los desarrolladores en el uso de las herramientas MDD

7R. El Project Manager del proyecto MDD deberá validar la cadena de herramientas

8R. La validación de la cadena de herramientas tendrá las características siguientes:

8Ra. Un desarrollador de la aplicación de negocios no deberá modificar nunca un artefacto MDD ya generado.

8Rb. Las herramientas, deberán estar totalmente integradas con el Sistema de Administración de Configuraciones (SC, Configuration Management System en inglés) definido. Deberá definirse la estrategia del Sistema de Administración de Configuraciones en la cadena de herramientas MDD.

9R. Deberá ser posible regenerar todos los artefactos de la aplicación de negocios en forma automática, a partir de un archivo generado para ese fin.

10R. Una vez construido el plan de proyecto el seguimiento y control del proyecto MDD no es diferente al de cualquier otro proyecto de desarrollo de software

11R. El éxito de un Proyecto MDD depende del éxito en la reutilización de los artefactos de los modelos:

11Ra. Es importante analizar el valor y contribución de un artefacto para determinar si debe o no ser generado. La administración de artefactos debe incluir:

11Ra1. La identificación y recuperación de un artefacto para su reuso

11Ra2. Asegurarse que se recupera el artefacto adecuado para la versión de ejecución que corresponde.

11Ra3. Chequear la integridad de un artefacto y verificar si la versión es la última y la apropiada.

11Ra4. Chequear la certificación de un artefacto y si está certificada para correr en el entorno correspondiente.

12R. El seguimiento de un proyecto MDD es similar a cualquier otro proyecto de desarrollo de software (10R). Pero hay ventajas adicionales que aporta el MDD:

12Ra. Cuando se realiza el seguimiento, del uso de las herramientas, durante el desarrollo de la aplicación, se generan reportes de status al mismo tiempo que se genera el código.

12Rb. Los casos de test generados, pueden escribirse de modo que guarden automáticamente los resultados de los test cada vez que se corren.

El resultado de estas dos prácticas es que se obtienen datos que reflejan en forma exacta el real progreso del proyecto. Esto permite tener en forma anticipada un aviso de potenciales desvíos dando la oportunidad de hacer correcciones.

13R. En el ciclo de vida de un proyecto MDD las herramientas desarrolladas deben crearse y testearse apropiadamente, para asegurarse que las soluciones que genera son exactas y consistentes.

El marco de trabajo cubre, la creación, testing y desarrollo de los modelos, patrones y transformaciones que generarán la solución de negocios.

13Ra. Una práctica usada comúnmente es que los expertos, manualmente o utilizando las herramientas comunes de la plataforma, creen una solución ejemplo.

13Rb. Las transformaciones deben ser probadas, antes de comenzar la generación y lanzamiento de la solución.

14R. Debe existir un mecanismo para el reemplazo, o el desarrollo de nuevas versiones que pueden coexistir, y asegurarse que sean accesibles por los usuarios adecuados.

15R. Debe determinarse el nivel de versionado (por archivo, por clase, por servicio, unidad de desarrollo y otros) a aplicar. Se versionan transformaciones, patrones, perfiles y todos los artefactos reusables.

16R. Se recomienda tener un mecanismo para certificar que los artefactos y modelos cumplan los estándares y se mantenga la integridad del sistema. A esta práctica se la llama Certificación de Servicio del modelo o artefacto.

4.2 Claudia Pons et al [1]

De acuerdo al texto “Desarrollo de software dirigido por Modelos” [1] de Claudia Pons et al, en el proceso MDD se deben realizar las siguientes tareas:

1P. La plataforma MDD debe ser desarrollada por los profesionales más experimentados, que son:

1Pa. Los expertos en el dominio,

1Pb. Los desarrolladores del lenguaje

1Pc. Los modeladores o ingenieros del PIM

1Pd. El ergonomista

1Pe. Los desarrolladores de las transformaciones y de los generadores de código

1Pf. Los expertos en el marco del dominio o ingenieros del Modelo Específico de la Plataforma.

2P. Es aconsejable separar el desarrollo en varias iteraciones.

3P. Se recomienda tener en cuenta las siguientes guías durante el desarrollo del proyecto:

3Pa. Realizar una inversión explícita en las herramientas de soporte

3Pb. Utilizar a la gente más calificada para desarrollar las herramientas MDD con el objetivo de capturar y automatizar su experiencia

3Pc. Considerar que además del código, el proyecto generará documentos, configuraciones, reportes y casos de prueba

3Pd. Asegurarse que el proceso de desarrollo soporta ambientes de prueba además de ambientes de producción

3Pe. Definir las estrategias de manejo de configuraciones para las herramientas MDD.

3Pf. Asignar tiempo al entrenamiento del equipo sobre el uso de herramientas MDD

3Pg. Destinar tiempo para considerar si las herramientas MDD serán reusables en proyectos futuros.

4P. Al finalizar el proyecto MDD, es útil generar las siguientes métricas:

4Pa. El costo de desarrollo de las herramientas MDD

4Pb. La productividad de los desarrolladores de la aplicación al usar las herramientas. Comparar con el esfuerzo que hubiera sido necesario para desarrollar todo el código manualmente.

4Pc. El nivel de calidad logrado por el equipo de desarrollo

4Pd. El esfuerzo requerido para lograr que las herramientas MDD puedan ser reutilizadas en otros proyectos.

5P. Identificar, desarrollar e instalar las herramientas MDD requeridas, antes que los desarrolladores de la aplicación de negocios las necesiten.

6P. La administración de los artefactos MDD, sus descripciones relacionadas y el mantenimiento de sus repositorios se torna un tema relevante.

4.3 Erkuden et al[3]

En este caso se han tomado la totalidad de las buenas prácticas, que corresponden al grado de madurez 5 definido por los autores.

De acuerdo a lo que indican Erkuden et al, una organización que califica para el nivel 5 cumple además, las prácticas de todos los niveles anteriores.

En el trabajo de Erkuden et al, se consideran tres tipos de prácticas, las de Ingeniería, las de Gerenciamiento del Proyecto y las de Soporte, identificadas con los prefijos ENG, PJM y SUP, respectivamente. Todas ellas se tendrán en cuenta.

- 1E-ENG1 Identificar las técnicas de modelado
- 2E-ENG2 Definir el modelo Técnico
- 3E-ENG3 Generar código a partir del modelo Técnico
- 4E-ENG4 Generar documentación a partir del modelo Técnico
- 5E-ENG5 Completar el código para cumplir todos los requerimientos
- 6E-PJM1 Decidir las herramientas de modelado
- 7E-ENG6 Definir el modelo de Negocios (Business Model)
- 8E-ENG7 Definir transformaciones para pasar del modelo Técnico a texto
- 9E-ENG8 Separar el código generado del no generado
- 10E-ENG9 Chequear los modelos
- 11E-PJM2 Definir el Workflow del proyecto MDD
- 12E-PJM3 Decidir la cobertura de las actividades de modelado
- 13E-SUP1 Establecer y mantener repositorios para los modelos y transformaciones
- 14E-SUP2 Definir, recoger y analizar medidas con respecto a las actividades de modelado
- 15E-ENG10 Definir el metamodelo centrado en la arquitectura
- 16E-ENG11 Definir el modelo de Dominio
- 17E-ENG12 Definir las transformaciones del modelo de negocios (Business model) al modelo técnico
- 18E-ENG13 Simular modelos
- 19E-ENG14 Separar los modelos técnicos del producto e infraestructura de la familia de sistemas.
- 20E-PJM4 Administrar el desarrollo de la infraestructura común
- 21E-ENG15 Definir lenguajes de dominio específicos
- 22E-ENG16 Mejorar y validar continuamente los metamodelos
- 23E-ENG17 Definir transformaciones del modelo de dominio al modelo de negocios
- 24E-ENG18 Validación y Verificación basadas en el modelo
- 25E-PJM5 Establecer y mantener los elementos MDD estratégicos

Haremos el análisis de las Prácticas Específicas de cada Area de Procesos del CMMI dev 1.3 [4], nivel 2, utilizando las 47 buenas prácticas seleccionadas.

5. Soporte de MDD a las Áreas de Proceso de Nivel 2 del CMMI-DEV 1.3

Se ha evaluado cada una de las prácticas específicas de cada Área de Proceso, de acuerdo a las enumeradas y descritas en el documento CMMI dev 1.3 [4] y se ha analizado una por una, para determinar si existen prácticas MDD que las satisfagan. Para ello, se busca que existan en MDD, actividades, artefactos, personas o

workflows, que las implementen. Por ejemplo, en el Área de Proceso, Gestión de Configuraciones, la práctica específica SP1.1 [4] expresa, Identificar los Ítems de Configuración. MDD brinda soporte a esta práctica específica porque, la práctica 5R, indica que deben producirse los artefactos de muestra para los escenarios clave, la práctica 6R, indica el proceso para definir la cadena de herramientas MDD, la práctica 8R asegura la integración de los elementos generados, la práctica 3P indica que deben definirse estrategias de manejo de configuraciones, la práctica 5P se refiere a la oportunidad en que deben generarse las herramientas MDD, las prácticas 1E-ENG2, 2E-ENG2, 3E-ENG3, 5E-ENG5, 6E-PJM1, 7E-ENG6, 8E-ENG7, 17E-ENG12 y 25E-PJM5 garantizan que serán identificados los Ítems de Configuración. Por lo que existen prácticas MDD que indican acciones, que al cumplirse satisfacen el objeto de la práctica CMMI. Analizando la práctica específica 1.3 de la misma área, que dice, Crear o Liberar Líneas de Base para Uso Interno y Entrega al Cliente, concluimos que está soportada por las prácticas MDD 2R, 4R, 5R y 6R que se refieren a la identificación de modelos MDD que puedan reusarse, a la identificación del modelo UML de componentes y a los pasos a seguir para definir la cadena de herramientas. Haciendo un análisis similar para cada una de las prácticas específicas del Area de Proceso, Gestión de Configuraciones, se concluye, que de las 7 prácticas específicas, MDD soporta 5, o sea el 71%.

Consideremos, también a modo de ejemplo, la práctica específica SP1.1 del Area de Proceso, Gestión de los Requerimientos, que expresa, Comprender los Requerimientos, podemos decir que la soportan, las prácticas MDD 1R a 6R, que apuntan a la comprensión de los requerimientos para luego construir el aplicativo de negocios. Las prácticas 1E-ENG 1, 5E-ENG 5, 6E-PJM 1, 12E-PJM 3 y 14E-SUP 2 indican los procedimientos a seguir para construir la cadena de herramientas MDD comprendiendo los requerimientos. La práctica específica SP 1.5 [4] de la misma área, dice, Asegurar el Alineamiento entre los Productos de Trabajo y los Requerimientos. En este caso, el soporte MDD se basa en las prácticas MDD, 5R, que indica que deben producirse artefactos de muestra para los escenarios clave, la práctica 6R que habla de la necesidad de validar la cadena de herramientas para garantizar el alineamiento de los productos de trabajo con los requerimientos. La aplicación de la práctica MDD 11R, asegura que se mantendrá la trazabilidad y alineamiento entre los productos de trabajo y los requerimientos. En este Area de Proceso, haciendo un análisis de las 5 prácticas específicas se obtiene soporte total del 100%.

Se ha considerado que un área de proceso tiene un alto soporte MDD cuando la cantidad de prácticas específicas que se cumplen, es por lo menos del 70%. Se describen brevemente a continuación los resultados obtenidos para las Areas de Proceso nivel 2 de CMMI dev 1.3.

5.1 Gestión de Configuraciones. Por definición de MDD, deben generarse desde el inicio del desarrollo de un proyecto ([1], [2] y [3]), los elementos que serán los identificados bajo este área. Son muchas las prácticas MDD que aplican, entre otras, las prácticas 5R, 6R, 7R, 11R, 12 R, 14R, 3P, 5P, 1E-ENG1, 2E-ENG5, 4E-ENG4, 13E-SUP1, 25E-PJM5. Este Area de Proceso posee 7 prácticas específicas de las que MDD soporta 5, que corresponden al 71% del total.

5.2 Gestión de los Acuerdos con Proveedores. MDD no soporta este área de proceso.

5.3 Gestión de los Requerimientos. En este caso, el soporte MDD es total, dado que hablar de requerimientos en un proyecto MDD, significa definir las características y administración de los artefactos MDD y los procedimientos para hacerlo, están detalladamente enunciados por todos los autores que tenemos como referencia [1], [2] y [3] en este trabajo. Para mencionar solamente algunas de las prácticas MDD que aplican, podemos enunciar las siguientes: 1R a 6R 13R, 14R y 16R, 1P, 3P y 6P, 1E-ENG1, 6E-PJM1, 5E-ENG 5, 12E-PJM3, 14E-SUP2. Este Área de Proceso posee 5 prácticas específicas, y MDD soporta todas, lo que significa el 100% del total.

5.4 Aseguramiento de la Calidad del Proceso y del Producto. Se ha verificado un alto soporte MDD a esta área de procesos a través de las siguientes prácticas: 5R, 6R, 11R, 13R, 16R, 3P, 6P, 10E-ENG9, 25E-PJM5. Este Área de Proceso posee 4 prácticas específicas, de las que MDD soporta 3, que corresponden al 75% del total.

5.5 Métricas y Análisis. En este caso, se ha encontrado una menor cantidad de prácticas MDD que aplican y son las siguientes: 4P, 6P, 9R, 11R y 13E-SUP1. Este Área de Proceso posee 8 prácticas específicas de las que MDD soporta 3, que corresponden al 37,5% del total.

5.6 Monitoreo y Control del Proyecto. Se ha encontrado un bajo soporte MDD en este caso y solamente se ha encontrado una práctica MDD que es aplicable que es la 12R. Este Área de Proceso posee 10 prácticas específicas y MDD soporta 1, que corresponde al 10% del total.

5.7 Planeamiento del Proyecto. En este caso el soporte MDD es alto y las prácticas MDD que aplican son, 1R a 6R, 8R, 1P, 2P, 3P, 6E-PJM1, 11E-PJM 2, 12E-PJM3, 20E-PJM4, 25E-PJM5. Este Área de Proceso posee 14 prácticas específicas de las que MDD soporta 10, que corresponden al 71% del total.

6. Discusión de los resultados y conclusiones.

La Tabla 1, muestra para cada área de proceso, el número de prácticas específicas definidas por CMMI-DEV 1.3, el número soportado por MDD y el porcentaje que representa. La Tabla 2, indica las prácticas específicas no soportadas por MDD.

Tabla 1. Áreas de proceso CMMI y soporte MDD

Área de proceso	Número de prácticas específicas	Número soportado por MDD	Porcentaje soportado por MDD
Gestión de Configuraciones	7	5	71%
Gestión de los Acuerdos con	6	0	0%

Proveedores			
Gestión de los Requerimientos	5	5	100%
Aseguramiento de la Calidad del Proceso y del Producto	4	3	75%
Medición y Análisis	8	3	37,5%
Monitoreo y Control del Proyecto	10	1	10%
Planeamiento del Proyecto	14	10	71%

Tabla 2. Prácticas específicas no soportadas en MDD, por área de proceso

Área de Proceso	Prácticas específicas no soportadas por MDD
Gestión de Configuraciones	Rastreo de los pedidos de cambio, Realizar auditorías de configuración.
Gestión de los Acuerdos con Proveedores	MDD no soporta ninguna de las prácticas específicas
Gestión de los Requerimientos	No hay
Aseguramiento de la Calidad del Proceso y del Producto	Comunicar y solucionar los problemas no resueltos.
Métricas y Análisis	Especificar Procedimientos de Análisis, Obtener datos de las mediciones, Analizar datos de las mediciones, Almacenar datos y resultados, Comunicar los resultados.
Monitoreo y Control del Proyecto	Compromiso del Monitoreo, Monitorear los Riesgos del Proyecto, Monitorear la Administración de los Datos, Monitorear la participación de los stakeholders, Conducir las revisiones del progreso, Conducir las revisiones de los hitos, Analizar los problemas, Tomar acciones correctivas, Administrar las acciones correctivas.
Planeamiento del Proyecto	Establecer el Presupuesto y Programación, Revisión de todos los Planes que afectan el Proyecto, Reconciliar Niveles de Trabajo y Recursos Necesarios, Obtener Compromiso con el Plan.

El Área de Proceso Gestión de los Acuerdos con Proveedores, no aplica a un proyecto MDD, ya que la subcontratación de productos y servicios externos forma parte de la estrategia de la organización y está fuera de su alcance.

En el caso del Área de Proceso Medición y Análisis, debería mejorarse el soporte MDD, enunciando e implementando prácticas que apunten a soportar las prácticas específicas no cubiertas.

El Área de Proceso Monitoreo y Control de Proyecto tiene muy bajo soporte MDD, y esto debe ser mejorado. Se debe, en parte, a que los autores en los que se basa la selección de buenas prácticas, afirman, que el seguimiento de un proyecto MDD es similar al de cualquier otro proyecto de software (práctica 10R, y Pons et al en el texto de la referencia) y no se han fijado prácticas específicas que apunten a la problemática del seguimiento y control del proyecto. Sin embargo, hay cuestiones que pueden ser analizadas en mayor detalle, como el manejo del riesgo y acciones

correctivas a lo largo del desarrollo, que pueden ser específicas del MDD. Otro tema que no se ha profundizado, se refiere a los costos de un proyecto MDD y como se van mitigando con el tiempo a medida que se construyen los repositorios y el reuso se hace factible. Si bien, existen prácticas MDD recomendando la evaluación del reuso de los artefactos al momento de ser diseñados y construidos (prácticas 11R, 3P, y 13E.-SUP1), no existen prácticas concretas que aseguren que esto se haga efectivo. Este punto débil del MDD se debe, entendemos, a que todavía no existe suficiente experiencia de uso y reuso de los artefactos MDD en las organizaciones. Pero debería evaluarse como debieran evolucionar los costos idealmente a lo largo del tiempo, en la medida en que una organización gane experiencia y construya su repositorio.

Hasta el momento, MDD se focaliza en el trabajo técnico y en este sentido se puede concluir que existen, a la fecha, muchas recomendaciones sobre la metodología y secuencia necesaria para el desarrollo de un proyecto MDD como se muestra en el alto grado de soporte en las Areas de Proceso, Gestión de Configuraciones, Gestión de los Requerimientos, Aseguramiento de la Calidad del Proceso y del Producto y Planeamiento del Proyecto.

Se han destacado en este trabajo, algunas cuestiones de importancia, que entendemos no han sido analizadas aún con precisión. Si bien, las Areas de Proceso y prácticas específicas no soportadas, o débilmente soportadas por MDD, se explican en gran parte, por la reciente irrupción de MDD en el desarrollo de software, marcan sin embargo, los problemas que deberían ir resolviéndose para estimular su uso en las organizaciones.

7. Referencias

1. Desarrollo dirigido por modelos, Claudia Pons, Roxana Giardini, Gabriela Perez. Edulp Editorial. Coedición con Mc Graw Hill Educación 2010, 279 páginas. ISBN: 978-950-34-0630-4. (www.editorial.unlp.edu.ar/libro_pons.html)
2. Patterns: Model-Driven Development Using IBM Rational Software Architect, Peter Swithinbank, Mandy Chessell, Tracy Gardner, Catherine Griffi n, Jessica Man, Helen Wylie, Larry Yusuf, disponible en ibm.com/redbooks
3. MDD Maturity Model: A Roadmap for Introducing Model-Driven Development, Erkuden Rios, Teodora Bozheva, Aitor Bediaga, Nathalie Guilloreau. A. Rensink and J. Warner(Eds): ECMDA-FA 2006, LNCS 4066, pp 78-89, 2006.
4. CMMI dev 1.3 (www.sei.cmu.edu/reports/10tr033.pdf).
5. www.omg.org
6. MDA Explained, The Model Driven Architecture : Practice and Promise. Anneke Kleppe, Jos Warner, Wim Bast, Addison-Wesley Professional; 1 edition (May 1, 2003).
7. www.sei.cmu.edu